

CIBERSEGURIDAD

'Bootcamp IX'



Informe Práctica Módulo Recopilación de Información.

Maximiliano Dariel Altamirano.

Academia KeepCoding.

INDICE

INFORME	3
Descripción de la práctica	3
Descripción del dominio	3
FOOTPRINTING	4
Análisis activo	4
Análisis pasivo	7
FINGERPRINTING	9
ANALISIS DE VULNERABILIDADES.....	17
OSINT.....	23
RESUMEN	29
Objetivo	29
Herramientas.....	29

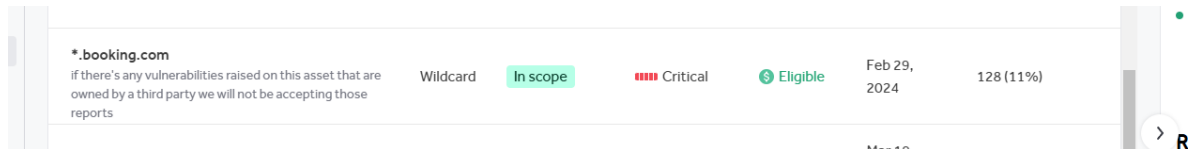


INFORME

Descripción de la práctica

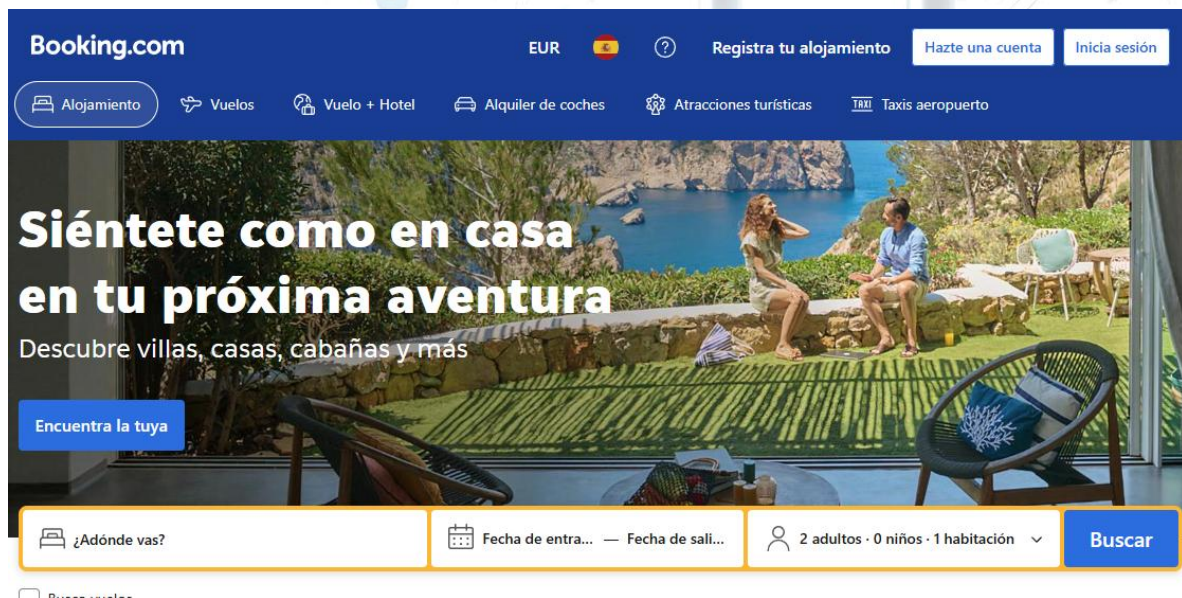
Para el desarrollo de la práctica realizaremos la recopilación de datos sobre el dominio **www.booking.com**, elegido por su scope amplio respetando las normas y condiciones de la plataforma **www.hackerone.com**.

Intentaremos identificar Bugs y Vulnerabilidades del dominio mencionado.



Descripción del dominio

Booking.com es una plataforma de reservas de alojamientos y tarifa de viajes, fue fundada en Países Bajos en 1996 y forma parte de Booking Holding Inc.



Abordaremos las diferentes etapas de la recopilación de datos con la siguiente estructura:

- Footprinting
- Fingerprinting
- Análisis de Vulnerabilidades
- OSINT

Como ejes principales de nuestro informe.

FOOTPRINTING

En este apartado intentaremos identificar los subdominios con reconocimiento vertical de nuestra Web objetivo, utilizando técnicas activas y pasivas.

Análisis activo

Utilizaremos **Shuffledns** para identificar los posibles subdominios de nuestra Web objetivo con fuerza bruta, lanzamos el siguiente comando:

```
shuffledns -mode bruteforce -d booking.com -w  
$HOME/recopilacion/lists/domains.txt -r  
$HOME/recopilacion/lists/resolvers.txt -silent > shuffledns.txt
```

En este primer informe identificamos 94 subdominios, de los cuales marcaremos como relevantes:

secure.booking.com: Este dominio podría estar relacionado con la seguridad y autenticación.

developer.booking.com: Podría proporcionar información sobre APIs y recursos para desarrolladores.

api.booking.com: Este dominio probablemente esté relacionado con las APIs de Booking.com.

admin.booking.com: Podría estar relacionado con la administración interna del sitio.

bugs.booking.com: Podría ser un dominio utilizado para reportar y rastrear errores, útil para conocer el procedimiento ante tipo de problemática.

jira.booking.com: Podría estar relacionado con el sistema de seguimiento de problemas y proyectos de Booking.com.

vpn.booking.com: Este dominio podría estar relacionado con la red privada virtual (VPN) de Booking.com.

Analizamos con **Analyticsrelationships** si nuestra Web objetivo tiene definido un ID de Google Analytics:

```
analyticsrelationships --url https://www.booking.com/
```

```
(kali@kali)-[~/recopilacion/booking.com/12022025]
$ analyticsrelationships --url https://www.booking.com/ > googleanalytics.txt
/usr/bin/analyticsrelationships:34: SyntaxWarning: invalid escape sequence '\d'
pattern = "UA-\d+--\d+"
/usr/bin/analyticsrelationships:47: SyntaxWarning: invalid escape sequence '\.'
pattern = "(www\.\googletagmanager\.com/ns\.html?id=[A-Z0-9\-\_]+)"
/usr/bin/analyticsrelationships:49: SyntaxWarning: invalid escape sequence '\d'
pattern3 = "UA-\d+--\d+"
/usr/bin/analyticsrelationships:78: SyntaxWarning: invalid escape sequence '\-'
pattern = "/relationships/[a-z0-9\-\_\.\-]+\.[a-z]+"

UA-ID
DOMAINS

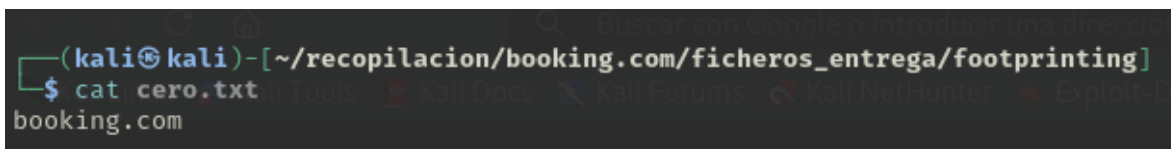
> Get related domains / subdomains by looking at Google Analytics IDs
> Python version
> By @JosueEncinar

[+] Analyzing url: https://www.booking.com/
[+] URL with UA: https://www.googletagmanager.com/gtm.js?id=GTM-5Q664QZ
[+] Obtaining information from builtwith and hackertarget
[-] Analytics ID not found...
```

En esta oportunidad nuestra Web objetivo no cuenta con ID para progresar en el análisis.

Exploramos los certificados SSL/TLS con la herramienta **Cero** intentando identificar los dominios y subdominios que puedan pertenecer a la organización objetivo relacionados a estos certificados. Lanzamos el siguiente comand:

```
cero -d booking.com | grep 'booking.com' > cero.txt
```



```
(kali@kali)-[~/recopilacion/booking.com/ficheros_entrega/footprinting]
$ cat cero.txt
booking.com
```

Nuestro análisis solo a identificado el dominio principal, sin datos que destacar.

Utilizamos la técnica de Web Scraping para extraer información de nuestra Web objetivo con la herramienta **Katana** y **Unfurl** para obtener los subdominios de la url. lanzaremos el siguiente comando:

```
echo booking.com | katana -jc -o katanaoutput.txt -kf robotstxt,
sitemapxml | unfurl --unique domains > katana.txt
```

Hemos identificado 19 subdominios, ninguno trascendente para destacar ya que algunos datos los hemos mencionado en el primer análisis.

Análisis pasivo

Revisaremos los certificados asociados a la Web objetivo con la intención de identificar los subdominios vinculados a estos logs, para ello utilizaremos la herramienta **CTFR** lanzando el siguiente comando:

```
ctfr -d booking.com | unfurl --unique domains > ctfr.txt
```

Encontramos 947 subdominios, de los cuales podemos destacar la siguiente información:

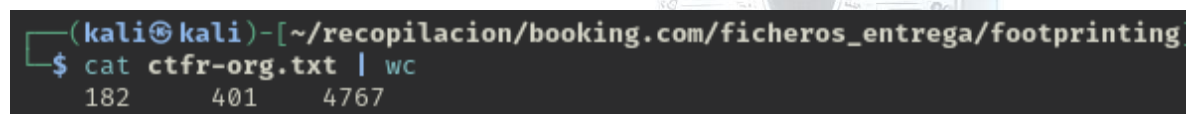
tls12-secure-distribution-xml.booking.com // tls12-secure-supply-xml.booking.com: Subdominios interesantes para evaluar configuraciones de TLS, certificados SSL y posibles vulnerabilidades.

ams4.dev.booking.com, *.dev.booking.com, dev.booking.com: Entonces de producción, suelen configurarse con menos seguridad que un entorno de producción.

gdpr.support.booking.com: Posiblemente relacionado con cumplimiento legal y protección de datos (GDPR).

Solo como dato adicional, sumaremos un fichero con los certificados asociados a la organización "Booking" modificando parámetros de la herramienta y lanzando el siguiente comando:

```
ctfr-org -d Booking > ctfr-org.txt
```

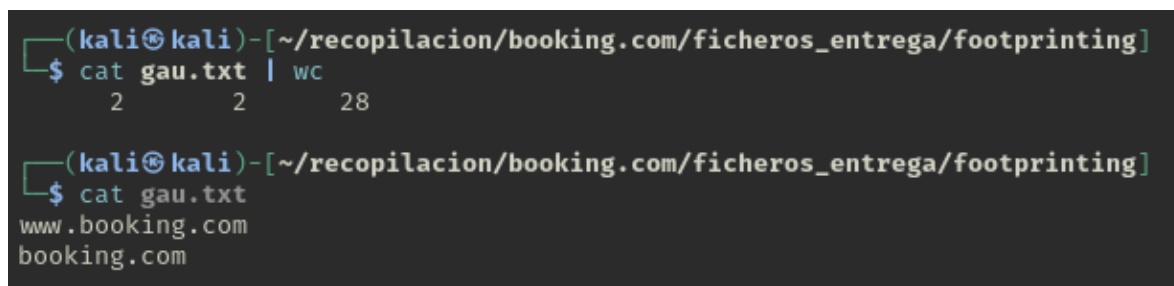


```
(kali@kali)-[~/recopilacion/booking.com/ficheros_entrega/footprinting]
$ cat ctfr-org.txt | wc
182      401     4767
```

Descubrimos 401 dominios y subdominios relacionados a la organización.

Haremos un reconocimiento sobre los archivos Web y Cache de nuestra Web Objetivo utilizando la herramienta **GAU**, lanzamos el siguiente comando:

```
gau --threads 5 booking.com --o gauoutput.txt --timeout 4 | unfurl  
--unique domains > gau.txt
```

A terminal window with a dark background. The prompt is (kali@kali)-[~/recopilacion/booking.com/ficheros_entrega/footprinting]. The first command is \$ cat gau.txt | wc, which outputs 2 2 28. The second command is \$ cat gau.txt, which outputs www.booking.com and booking.com.

```
(kali@kali)-[~/recopilacion/booking.com/ficheros_entrega/footprinting]  
$ cat gau.txt | wc  
2      2      28  
  
(kali@kali)-[~/recopilacion/booking.com/ficheros_entrega/footprinting]  
$ cat gau.txt  
www.booking.com  
booking.com
```

Para este apartado solo hemos identificado 2 dominios luego de 4hs de análisis de la herramienta, sin datos que destacar.

Unificamos todos los subdominios en un solo fichero denominado `presubdominios.txt`:

```
cat cero.txt ctfr.txt gau.txt katana.txt shuffledns.txt >  
presubdominios.txt
```

Con la información obtenida haremos permutaciones a los subdominios recolectados. También identificaremos de la nueva lista que subdominios se encuentran activos, por lo que utilizaremos las herramientas **ALTERX** y **DNSX** para contar con una base funcional. Lanzaremos el siguiente comando:

```
cat subdominios.txt | alterx | dnsx > alterx.txt
```

Para finalizar nuestro Footprinting vamos a concatenar toda la información recolectada con diferentes técnicas en el fichero denominado `subdominios.txt`

```
cat presubdominios.txt alterx.txt > subdominios.txt
```

En resumen, utilizaremos el fichero **"subdominios.txt"** como referencia, con sus 1261 subdominios encontrados, como parámetros en los siguientes análisis.

FINGERPRINTING

Para este apartado intentaremos descubrir información específica de nuestra Web objetivo, como tecnologías, puertos que utilizan sus servidores, sistema operativo, posibles waf entre otros. Para ello implementaremos varias herramientas automatizadas y también haremos recolección manual de información.

Gestionaremos con la herramienta **HTTPX** para identificar los dominios que se encuentran “online” en nuestra base, lanzaremos el siguiente comando:

```
cat subdominios.txt | httpx -silent | unfurl --unique domains > subdominiosfinal.txt
```

Así logramos identificar 291 subdominios online, sumaremos al listado de hallazgos el siguiente subdominio:

account.booking.com: Este subdominio podría ser de administración y gestión de cuentas.

Transformaremos nuestra base de subdominios DNS a fichero de IPs para realizar el scanner con la herramienta Masscan:

```
for subdominio in $(cat subdominiosfinal.txt); do dig +short $subdominio | grep -Eo '([0-9]{1,3}.){3}[0-9]{1,3}'; done > subdominiosfinal_ips.txt
```

Antes lanzaremos **Nmap** solo sobre el dominio principal para identificar los puertos abiertos y optimizar el scanner de Masscan con el siguiente comando:

```
sudo nmap -Pn -sS -sV -p0- booking.com > nmap.txt
```

Ya identificando los puertos abiertos intentaremos configurar los parámetros de manera más eficiente:

PORT	STATE	SERVICE	VERSION
80/tcp	open	http	Amazon CloudFront httpd
443/tcp	open	ssl/http	Amazon CloudFront httpd

Ocupamos **Masscan** y sumaremos, a los parámetros ya encontrados, los puertos más importantes para intentar identificar el acceso a puertos críticos:

```
sudo masscan -p22,25,53,80,443,3389,3306,1433,3389,8443 -iL  
subdominiosfinal_ips.txt > masscan.txt
```

En los resultados obtenidos no encontramos información trascendente más que los esperados, ya que el análisis muestra de manera redundantes los puertos 443 y 80.

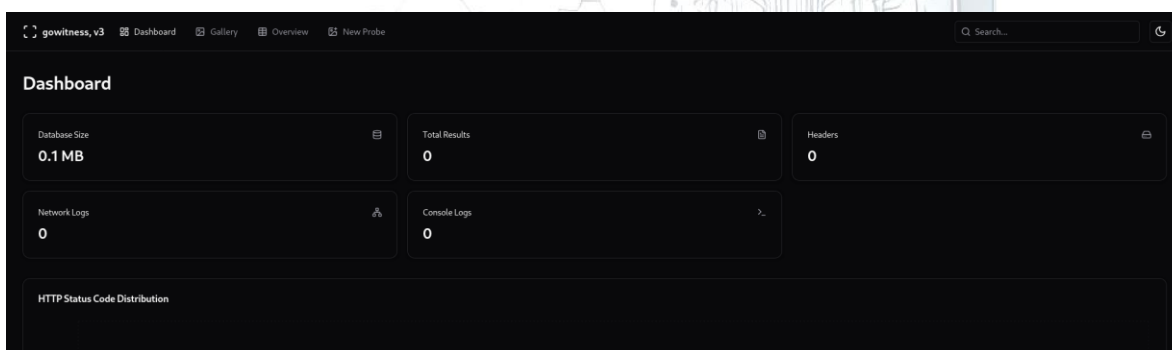
Utilizaremos la herramienta GoWitness para realizar capturas de pantalla de los subdominios encontrados con el siguiente comando:

```
gowitness scan file -f subdominiosfinal.txt
```

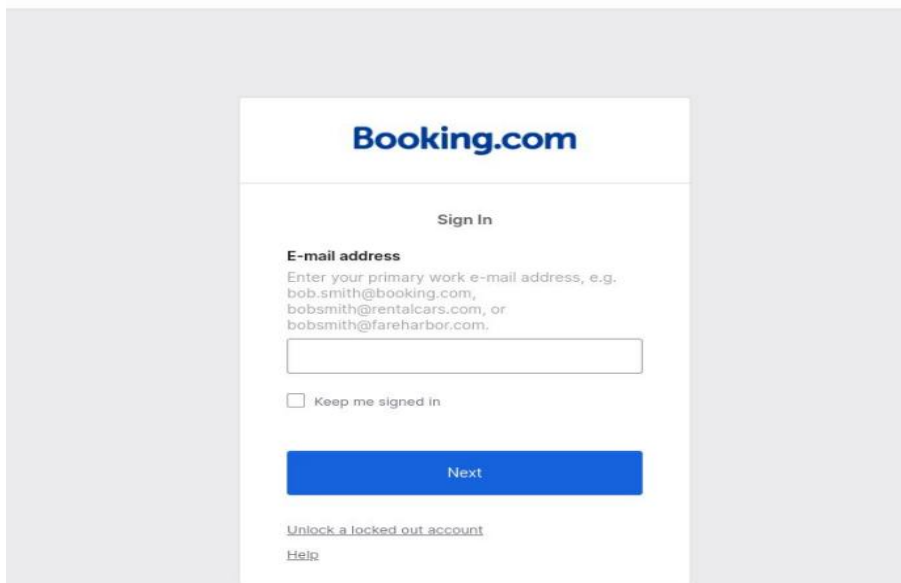
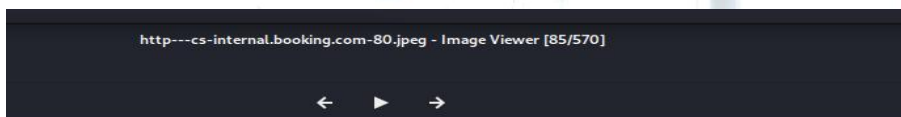
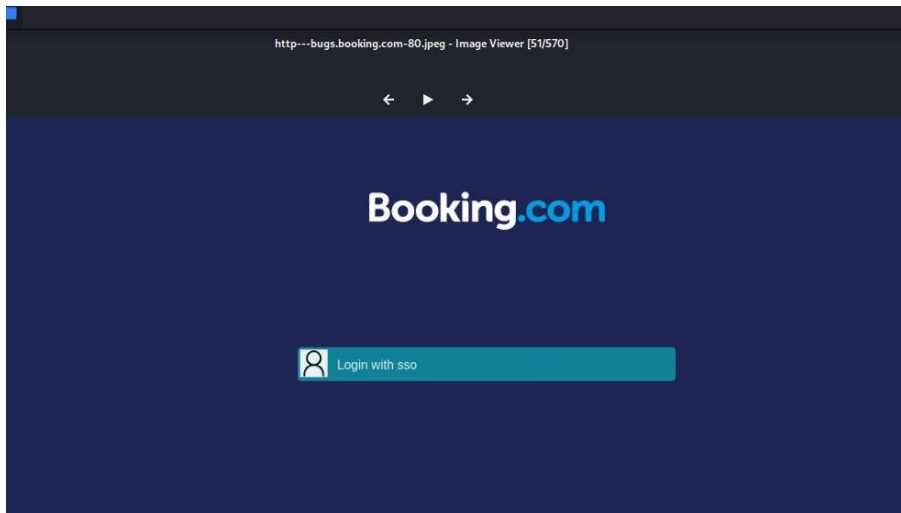
Aprovecharemos la interfaz de la herramienta para revisar los resultados desde, utilizamos el comando

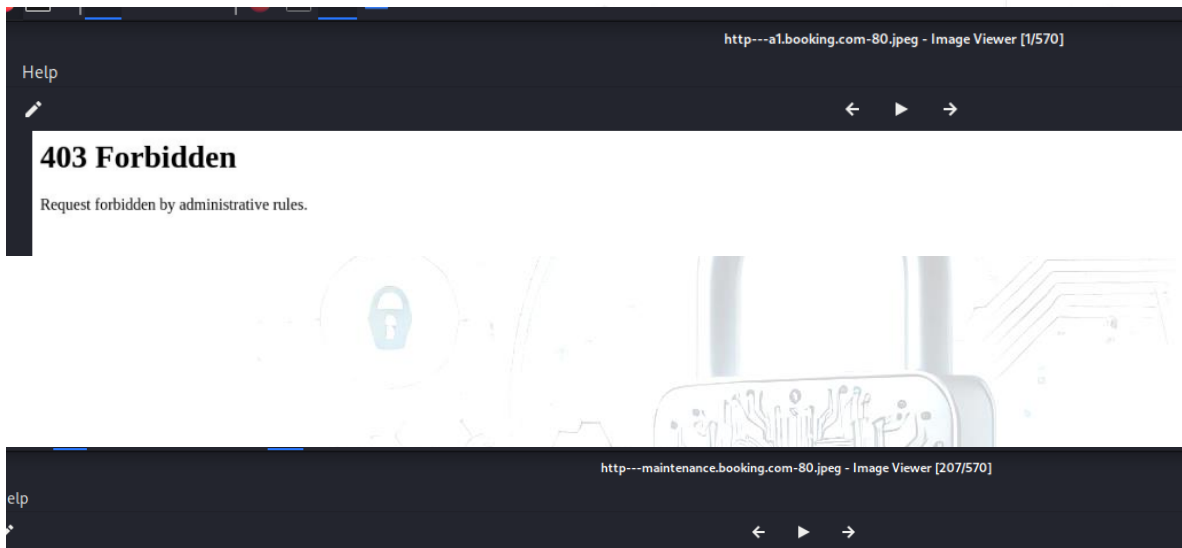
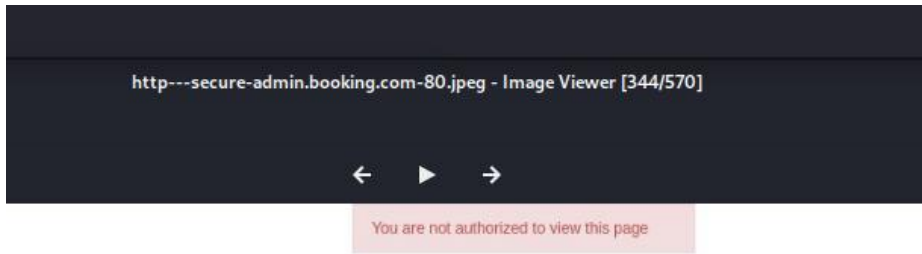
```
gowitness report server
```

Y accedemos al servidor <http://localhost:7171/>



Al no disponer de información la interfaz de la herramienta (sin actualización en la herramienta), haremos una exploración manual sobre las capturas de pantalla recolectadas en busca de indicios de vulnerabilidades.





403 ERROR

The request could not be satisfied.

Request blocked. We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner. If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.

Generated by cloudfront (CloudFront)
Request ID: nNwOdyjd6fr57u_keeYqbD1C_sUadmZcNu0E0NGo7C73YaNDi7CQA==

El error puede que visualizamos puede deberse a un tráfico excesivo o a una configuración incorrecta, ya que es recurrente el error en nuestras evidencias.

Sumaremos información con **Wappalyzer** de las tecnologías que utiliza nuestra Web objetivo.

The screenshot displays the Wappalyzer web application interface. At the top, there is a purple header with the Wappalyzer logo and navigation icons. Below the header, there are two tabs: "TECNOLOGÍAS" (selected) and "MÁS INFORMACIÓN". An "Export" button is located in the top right corner. The main content area is divided into two columns, each with a category header and a list of technologies.

Category	Technology	
Widget	Booking.com widget	
Analítica	Naver Analytics	
	Google Analytics	
Red de Publicidad	Google Publisher Tag	
	Google AdSense	
Tag Manager	Google Tag Manager	
Seguridad	DataDome	
	HSTS	
Miscelánea	Webpack	
	Open Graph	
	Module Federation	
Servidor Web	Nginx	
Librerías JavaScript	PubSubJS	
	Lodash 4.17.21	
jQuery Migrate	jQuery Migrate	
	jQuery 1.11.3	
PaaS	Amazon Web Services	
Proxy reverso	Nginx	

Framework Móvil



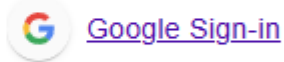
Cookie compliance



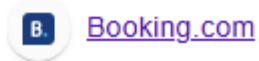
CDN



Autenticación



Marketing de afiliación



Destacaremos las siguientes tecnologías utilizadas:

Fortaleza:

DataDome: Generalmente usado para protección contra bots maliciosos, mitigación de ataques de scraping y protección frente a tráfico automatizado.

HSTS (HTTP Strict Transport Security): Añade una capa de seguridad al forzar el uso de conexiones HTTPS y prevenir ataques como "SSL stripping".


Oportunidad de mejora:

Webpack: Un empaquetador de módulos JavaScript, podría tener dependencias que necesitan auditorías regulares para identificar vulnerabilidades.

Identificaremos si nuestro objetivo está detrás de un Waf utilizando la herramienta **Waf00f** lanzando el siguiente comando.

```
wafw00f booking.com > wafw00f.txt
```

```
(kali@kali)-[~/recopilacion/booking.com/19022025]
$ cat wafw00f.txt
```



404 Hack Not Found
405 Not Allowed
403 Forbidden
502 Bad Gateway
500 Internal Error

~ WAFW00F : v2.3.1 ~
The Web Application Firewall Fingerprinting Toolkit

```
[*] Checking https://booking.com
[+] The site https://booking.com is behind Cloudfront (Amazon) WAF.
[~] Number of requests: 2
```

Podemos visualizar que **Cloudfront (Amazon)** es el Waf elegido por Booking para su dominio principal.

Haremos en scanner a nuestro fichero de subdominios con **Waf00f** para identificar más información sobre los Waf que utiliza la organización.

```
[~] Number of requests: 3
[*] Checking https://grafana.BOOKING.COM
[+] The site https://grafana.BOOKING.COM is behind Envoy (EnvoyProxy) WAF.

[~] Number of requests: 2
[*] Checking https://globalguru.booking.com
[+] The site https://globalguru.booking.com is behind Cloudflare (Cloudflare Inc.) WAF.
[~] Number of requests: 2

[*] Checking https://meetingrooms.booking.com
[+] The site https://meetingrooms.booking.com is behind Azure Front Door (Microsoft) WAF.
[~] Number of requests: 2
[*] Checking https://partner.BOOKING.COM
[+] The site https://partner.BOOKING.COM is behind CacheWall (Varnish) WAF.
```

Identificamos otros servicios Waf como Azure Front Door, Envoy, Cloudflare y CacheWall.

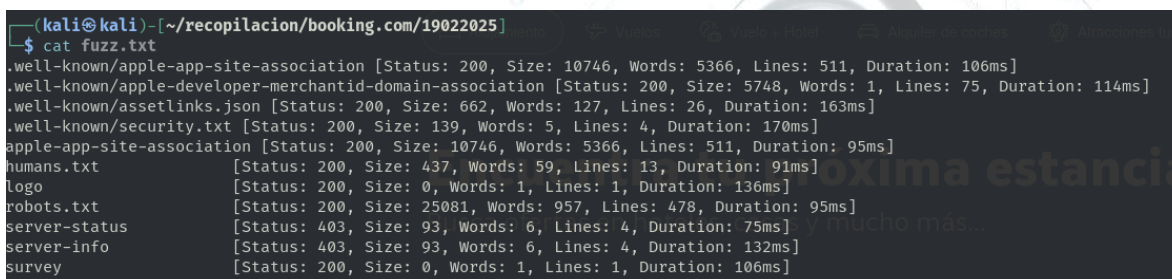
Intentaremos realizar un descubrimiento de contenidos utilizando la herramienta **ffuf**, lanzaremos el siguiente comando configurando los parámetros necesarios. En primera instancia utilizaremos la lista de palabras `common.txt` descargada del siguiente repositorio:

```
get
https://raw.githubusercontent.com/danielmiessler/SecLists/refs/heads/master/Discovery/Web-Content/common.txt
```

Ya con el listado completo, lanzamos el siguiente comando:

```
ffuf -w ~/recopilacion/lists/common.txt -t 10 -mc 200,401,403 -u
https://booking.com/FUZZ
```

Varios de los directorios descubiertos pueden ser objetivo de fuerza bruta.

A terminal window on a Kali Linux system showing the output of a ffuf scan on booking.com. The prompt is (kali@kali) - [~/recopilacion/booking.com/19022025]. The command executed is cat fuzz.txt. The output lists several discovered paths with their status, size, word count, line count, and duration. The paths include .well-known/apple-app-site-association, .well-known/apple-developer-merchantid-domain-association, .well-known/assetlinks.json, .well-known/security.txt, apple-app-site-association, humans.txt, logo, robots.txt, server-status, server-info, and survey.

```
(kali@kali) - [~/recopilacion/booking.com/19022025]
$ cat fuzz.txt
.well-known/apple-app-site-association [Status: 200, Size: 10746, Words: 5366, Lines: 511, Duration: 106ms]
.well-known/apple-developer-merchantid-domain-association [Status: 200, Size: 5748, Words: 1, Lines: 75, Duration: 114ms]
.well-known/assetlinks.json [Status: 200, Size: 662, Words: 127, Lines: 26, Duration: 163ms]
.well-known/security.txt [Status: 200, Size: 139, Words: 5, Lines: 4, Duration: 170ms]
apple-app-site-association [Status: 200, Size: 10746, Words: 5366, Lines: 511, Duration: 95ms]
humans.txt [Status: 200, Size: 437, Words: 59, Lines: 13, Duration: 91ms]
logo [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 136ms]
robots.txt [Status: 200, Size: 25081, Words: 957, Lines: 478, Duration: 95ms]
server-status [Status: 403, Size: 93, Words: 6, Lines: 4, Duration: 75ms]
server-info [Status: 403, Size: 93, Words: 6, Lines: 4, Duration: 132ms]
survey [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 106ms]
```

En resumen, nuestra Web objetivo implementa tecnologías y puertos configurados de forma correcta, gestionando con diferente Waf de confianza.

ANALISIS DE VULNERABILIDADES

Para este apartado utilizaremos diferentes herramientas de análisis, comenzaremos con **Greenbone**, gestionaremos la información de manera local desde el navegador.

Date ▼
Tue, Feb 18, 2025 11:04 AM UTC

Status
Done

Task
Booking

Severity
2.5 (Low)

High
0

Medium
0

Low
3

Log
150

False Pos.
0

Information

Results
(3 of 153)

Hosts
(3 of 4)

Ports
(0 of 2)

Applications
(2 of 2)

Operating Systems
(0 of 0)

CVEs
(0 of 0)

Closed CVEs
(0 of 0)

TLS Certificates
(0 of 0)

Error Messages
(14 of 14)

User Tags
(0)

Task Name
Booking

Scan Time
Tue, Feb 18, 2025 11:04 AM UTC - Tue, Feb 18, 2025 12:32 PM UTC

Scan Duration
1:27 h

Scan Status
Done

Hosts scanned
4

Filter
apply_overrides=0 levels=hml min_qod=70

Timezone
Coordinated Universal Time (UTC)

Information

Results
(3 of 153)

Hosts
(3 of 4)

Ports
(0 of 2)

Applications
(2 of 2)

Operating Systems
(0 of 0)

CVEs
(0 of 0)

Closed CVEs
(0 of 0)

TLS Certificates
(0 of 0)

Error Messages
(14 of 14)

User Tags
(0)

Vulnerability

Severity ▼

QoD

Host IP

Name

Location

Created

TCP Timestamps Information Disclosure

2.5 (Low)

80 %

3.160.231.75

server-3-160-231-75.mad53.r.cloudfront.n...

general/tcp

Tue, Feb 18, 2025 11:39 AM UTC

TCP Timestamps Information Disclosure

2.5 (Low)

80 %

3.160.231.72

server-3-160-231-72.mad53.r.cloudfront.n...

general/tcp

Tue, Feb 18, 2025 11:39 AM UTC

TCP Timestamps Information Disclosure

2.5 (Low)

80 %

3.160.231.53

server-3-160-231-53.mad53.r.cloudfront.n...

general/tcp

Tue, Feb 18, 2025 11:38 AM UTC

Applied filter: apply_overrides=0 levels=hml min_qod=70 find=1 sort=reverse=severity

Name

Oldest Result

Newest Result

Severity ▼

QoD

Results

Hosts

TCP Timestamps Information Disclosure

Tue, Feb 18, 2025 11:38 AM UTC

Tue, Feb 18, 2025 11:39 AM UTC

2.5 (Low)

80 %

3

3

Traceroute

Tue, Feb 18, 2025 11:38 AM UTC

Tue, Feb 18, 2025 11:39 AM UTC

8.0 (Low)

80 %

4

4

HTTP Server type and version

Tue, Feb 18, 2025 11:40 AM UTC

Tue, Feb 18, 2025 11:43 AM UTC

8.0 (Low)

80 %

16

4

HTTP Server Banner Enumeration

Tue, Feb 18, 2025 11:43 AM UTC

Tue, Feb 18, 2025 11:51 AM UTC

8.0 (Low)

80 %

10

4

HTTP TRACE Method Enabled

Tue, Feb 18, 2025 11:41 AM UTC

Tue, Feb 18, 2025 11:45 AM UTC

8.0 (Low)

70 %

12

4

HTTP Security Headers Detection

Tue, Feb 18, 2025 11:49 AM UTC

Tue, Feb 18, 2025 11:54 AM UTC

8.0 (Low)

80 %

16

4

Response Time / No 404 Error Code Check

Tue, Feb 18, 2025 11:14 AM UTC

Tue, Feb 18, 2025 11:15 AM UTC

8.0 (Low)

80 %

8

4

Web Application Scanning Consolidation / Info Reporting

Tue, Feb 18, 2025 11:57 AM UTC

Tue, Feb 18, 2025 12:00 PM UTC

8.0 (Low)

80 %

16

4

Hidden WWW Server Name (HTTP)

Tue, Feb 18, 2025 11:41 AM UTC

Tue, Feb 18, 2025 11:43 AM UTC

8.0 (Low)

70 %

4

4

nginx Detection Consolidation

Tue, Feb 18, 2025 11:34 AM UTC

Tue, Feb 18, 2025 11:35 AM UTC

8.0 (Low)

80 %

4

4

Applied filter: min_qod=70 sort=reverse=severity find=1 source=101

Revisando los resultados obtenidos, podemos concluir que nuestra Web objetivo tiene un grado de severidad **Bajo** a vulnerabilidades críticas conocidas, dificultando el acceso a terceros.

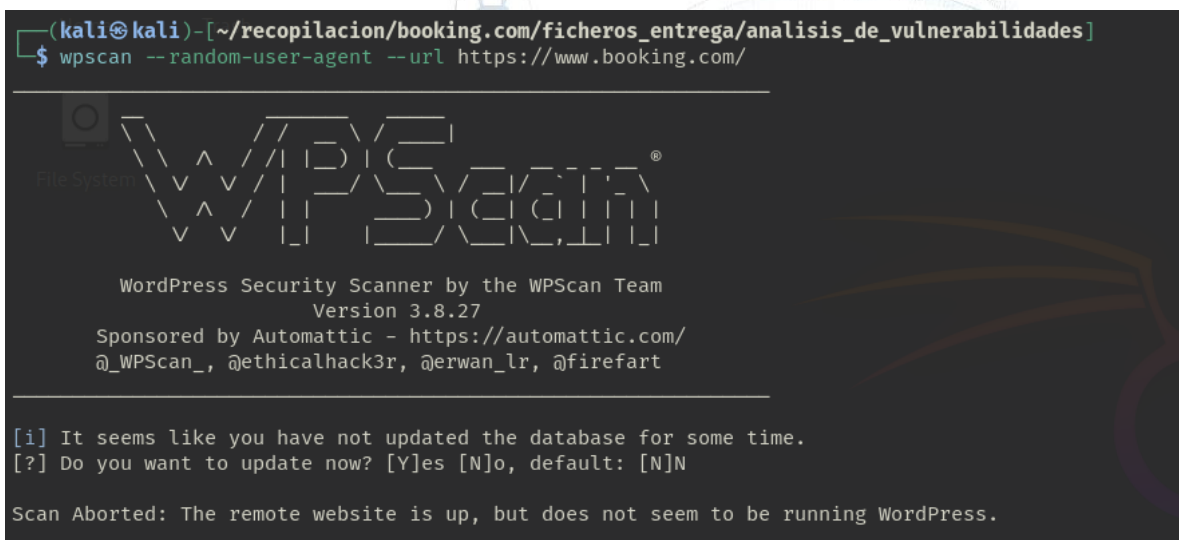
Realizamos un nuevo análisis con la herramienta **Nuclei**, lanzamos el siguiente comando:

```
nuclei -u booking.com > nuclei.txt
```

Luego del análisis y al revisar la información, podemos afirmar que las configuraciones y tecnologías implementadas son seguras.

Utilizaremos la herramienta WPScan para identificar si se utiliza WordPress. Lanzaremos el siguiente comando:

```
wpscan --random-user-agent --url https://www.booking.com/
```



```
(kali㉿kali)-[~/recopilacion/booking.com/ficheros_entrega/analisis_de_vulnerabilidades]
$ wpscan --random-user-agent --url https://www.booking.com/

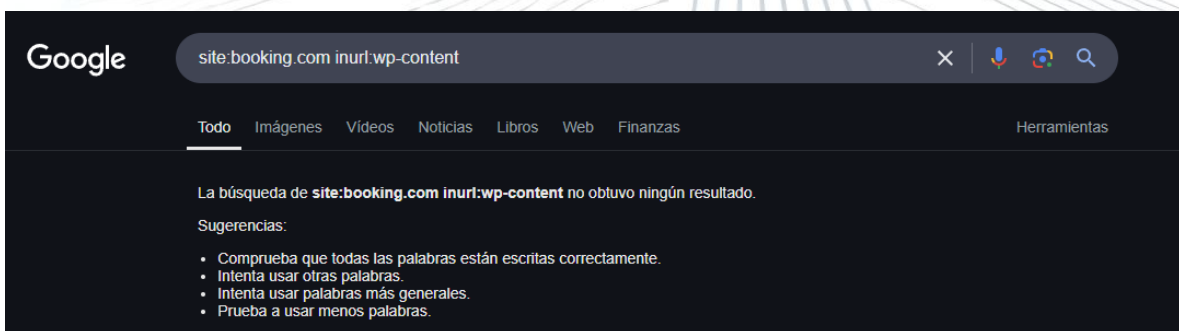
File System
  WPScan®

WordPress Security Scanner by the WPScan Team
Version 3.8.27
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

[i] It seems like you have not updated the database for some time.
[?] Do you want to update now? [Y]es [N]o, default: [N]N


Scan Aborted: The remote website is up, but does not seem to be running WordPress.
```

Confirmamos que no corre WordPress con una simple búsqueda en el navegador “site:booking.com inurl:wp-content”



Analizaremos con **Qualys** los certificados TLS/SSL de nuestra Web objetivo:

<https://www.ssllabs.com/ssltest/analyze.html?d=booking.com>

 **Qualys. SSL Labs**

Home Projects Qualys Free Trial Contact

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > booking.com

SSL Report: booking.com
Assessed on: Sat, 22 Feb 2025 07:16:49 UTC | [Clear cache](#)

[Scan Another >>](#)

	Server	Test time	Grade
1	18.239.199.111 server-18-239-199-111.sfo53.r.cloudfront.net Ready	Sat, 22 Feb 2025 07:05:20 UTC Duration: 169.288 sec	A+
2	18.239.199.14 server-18-239-199-14.sfo53.r.cloudfront.net Ready	Sat, 22 Feb 2025 07:08:09 UTC Duration: 166.989 sec	A+
3	18.239.199.21 server-18-239-199-21.sfo53.r.cloudfront.net Ready	Sat, 22 Feb 2025 07:10:57 UTC Duration: 181.874 sec	A+
4	18.239.199.64 server-18-239-199-64.sfo53.r.cloudfront.net Ready	Sat, 22 Feb 2025 07:13:59 UTC Duration: 170.697 sec	A+

SSL Report v2.3.1

También haremos las comprobaciones por consola con la herramienta **testssl** lanzando el siguiente comando:

```
./testssl.sh booking.com > testssl.txt
```

Del informe generado podemos destacar una potencial vulnerabilidad:

LUCKY13 (CVE-2013-0169): Potencialmente vulnerable, se utilizan cifrados CBC (Cipher Block Chaining) con TLS. Se recomienda revisar parches de seguridad.

```
Testing vulnerabilities
Heartbleed (CVE-2014-0160) not vulnerable (OK), no heartbeat extension
CCS (CVE-2014-0224) not vulnerable (OK)
Ticketbleed (CVE-2016-9244), experiment. not vulnerable (OK), reply empty
ROBOT not vulnerable (OK)
Secure Renegotiation (RFC 5746) supported (OK)
Secure Client-Initiated Renegotiation not vulnerable (OK)
CRIME, TLS (CVE-2012-4929) not vulnerable (OK)
BREACH (CVE-2013-3587) no gzip/deflate/compress/br HTTP compression (OK) - only supplied "/" tested
POODLE, SSL (CVE-2014-3566) not vulnerable (OK), no SSLv3 support
TLS_FALLBACK_SCSV (RFC 7507) No fallback possible (OK), no protocol below TLS 1.2 offered
SWEET32 (CVE-2016-2183, CVE-2016-6329) not vulnerable (OK)
FREAK (CVE-2015-0204) not vulnerable (OK)
DROWN (CVE-2016-0800, CVE-2016-0703) not vulnerable on this host and port (OK)
make sure you don't use this certificate elsewhere with SSLv2 enabled services, see
https://search.censys.io/search?resource=hosts&virtual_hosts=INCLUDE&q=2C29F8FC457421449E088FDEB9AD726732617CBCCAEFC99A88DD98573C25E5
LOGJAM (CVE-2015-4000), experimental not vulnerable (OK): no DH EXPORT ciphers, no DH key detected with <= TLS 1.2
BEAST (CVE-2011-3389) not vulnerable (OK), no SSL3 or TLS1
LUCKY13 (CVE-2013-0169), experimental not vulnerable (OK), uses cipher block chaining (CBC) ciphers with TLS. Check patches
Winshock (CVE-2014-6321), experimental not vulnerable (OK)
RC4 (CVE-2013-2566, CVE-2015-2808) no RC4 ciphers detected (OK)
```

Ejecutamos un análisis sobre los servicios de correo electrónico desde la siguiente web <https://dmarcian.com/>, logrando los siguientes resultados:

✓ DMARC

Your domain has a valid DMARC record and your DMARC policy will prevent abuse of your domain by phishers and spammers.

— Details

```
v=DMARC1; p=reject; rua=mailto:dmarc_rua@emaildefense.proofpoint.com,mailto:booking@dmarc.postmastery.eu;
ruf=mailto:dmarc_ruf@emaildefense.proofpoint.com
```

To understand and fix the specific errors, use our [DMARC Inspector](#).

✓ SPF

Great job! You have a valid SPF record, which specifies a hard fail (-all).

— Details

```
v=spf1 include:%{ir}%.%{v}%.%{d}.spf.has.pphosted.com -all
```

To understand and fix the specific errors, use our [SPF Surveyor](#).

La misma no ha encontrado registros DKIM para la organización analizada.

✗ DKIM

We couldn't find any DKIM records often associated with popular email sending sources. If you know the [specific selector](#), you can do a targeted search.

Enter selector

INSPECT DKIM

Podemos comprobar que el dominio principal utiliza registros validos de DMARC y SPF.

Replicaremos las comprobaciones desde consola utilizando la herramienta **spoofcheck.py**, lanzando el siguiente comando:

- `python spoofcheck.py booking.com > spoofcheck.txt`

```
(kali@kali)~/spoofcheck
$ python spoofcheck.py booking.com > spoofcheck.txt

(kali@kali)~/spoofcheck
$ cat spoofcheck.txt
[*] Found SPF record:
[*] v=spf1 include:%{ir}.$%{v}.$%{d}.spf.has.pphosted.com -all
[*] SPF record contains an All item: -all
[*] Found DMARC record:
[*] v=DMARC1; p=reject; rua=mailto:dmARC_rua@emaildefense.proofpoint.com,mailto:booking@dmARC.postmastery.eu; ruf=mailto:dmARC_ruf@emaildefense.proofpoint.com
[-] DMARC policy set to reject
[*] Aggregate reports will be sent: mailto:dmARC_rua@emaildefense.proofpoint.com,mailto:booking@dmARC.postmastery.eu
[*] Forensics reports will be sent: mailto:dmARC_ruf@emaildefense.proofpoint.com
[-] Spoofing not possible for booking.com
```

Con estos datos, validamos que no es posible hacer Spoofing sobre nuestra Web objetivo.

Para finalizar nuestro análisis de vulnerabilidades, vamos a explorar los posibles Subdomain Takeover con la herramienta **Subzy**. Lanzaremos el siguiente comando:

- `subzy run -targets subdominiosfinal.txt > subzy.txt` las siguientes vulnerabilidades:

```
(kali@kali)~/recopilacion/booking.com/ficheros_entrega/analisis_de_vuln
$ cat subzy.txt | grep -i "cargo collective"
[ VULNERABLE ] - admin.api.booking.com [ Cargo Collective ]
[ VULNERABLE ] - api.BOOKING.COM [ Cargo Collective ]
[ VULNERABLE ] - api.Booking.com [ Cargo Collective ]
[ VULNERABLE ] - api.booking.com [ Cargo Collective ]
[ VULNERABLE ] - ch.booking.com [ Cargo Collective ]
[ VULNERABLE ] - clicks.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw1gp4.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw1gp2.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw1gp5.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw1gp3.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw2gp5.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw2gp4.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw2gp2.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw2gp6.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw3gp3.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw3gp2.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw3gp4.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw6gp2.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw4gp3.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw4gp4.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw6gp4.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw5gp3.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw5gp4.booking.com [ Cargo Collective ]
[ VULNERABLE ] - gw6gp3.booking.com [ Cargo Collective ]
[ VULNERABLE ] - link.sg.booking.com [ Cargo Collective ]
[ VULNERABLE ] - mobile.api.booking.com [ Cargo Collective ]
[ VULNERABLE ] - pulse.api.booking.com [ Cargo Collective ]
[ VULNERABLE ] - s.sg.booking.com [ Cargo Collective ] 38.192.32
[ VULNERABLE ] - taxi.sg.booking.com [ Cargo Collective ] 38.192.32 sto53.r
[ VULNERABLE ] - www.relocatewithbooking.com [ Cargo Collective ]
```

De los resultados obtenidos podemos destacar dos posibles vulnerabilidades potenciales:

[DISCUSSION] - [Issue #152](https://github.com/EdOverflow/can-i-take-over-xyz/issues/152)

[DISCUSSION] - [Issue #22](https://github.com/EdOverflow/can-i-take-over-xyz/issues/22)

```
[ DISCUSSION ] - [Issue #152](https://github.com/EdOverflow/can-i-take-over-xyz/issues/152)
[ DISCUSSION ] - [Issue #152](https://github.com/EdOverflow/can-i-take-over-xyz/issues/152)
[ DISCUSSION ] - [Issue #22](https://github.com/EdOverflow/can-i-take-over-xyz/issues/22)
[ DISCUSSION ] - [Issue #152](https://github.com/EdOverflow/can-i-take-over-xyz/issues/152)
[ DISCUSSION ] - [Issue #152](https://github.com/EdOverflow/can-i-take-over-xyz/issues/152)
```

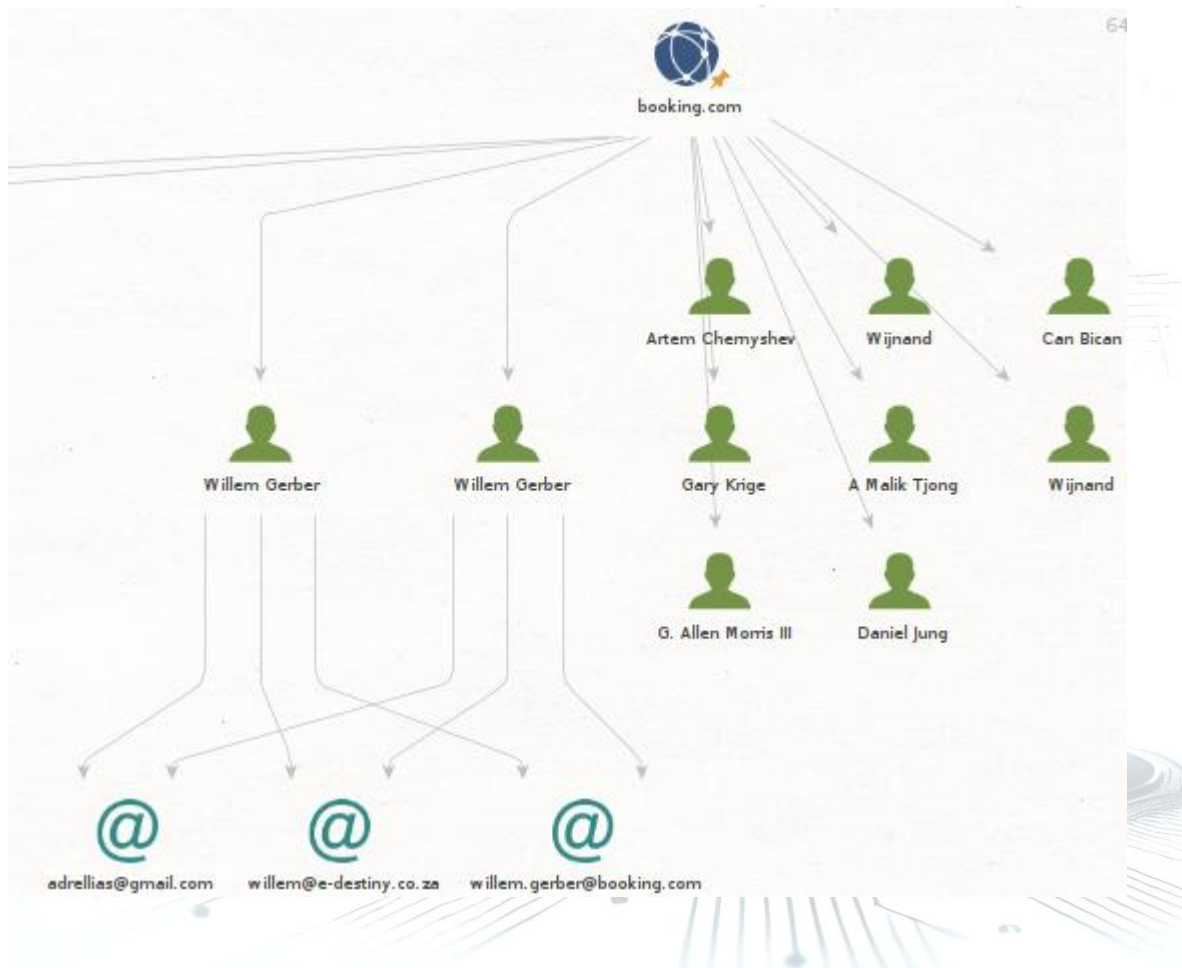
Resumiendo, luego del análisis de realizado con diversas técnicas y herramientas podemos concluir que nuestra Web objetivo utiliza protocolos de seguridad altos y tecnologías seguras, siento un objetivo difícil de atacar también en este apartado.



OSINT

Respecto a este punto haremos una recopilación de datos de fuentes públicas, utilizaremos navegadores y técnicas automatizadas para la práctica.

Lanzamos un análisis de la organización con la herramienta **Maltego**, en primera instancia haremos transformaciones buscando identificar personas relacionadas al dominio, posibles empleados. Utilizaremos el módulo "Email Addresses from Person" luego de identificar personas vinculada con la organización.

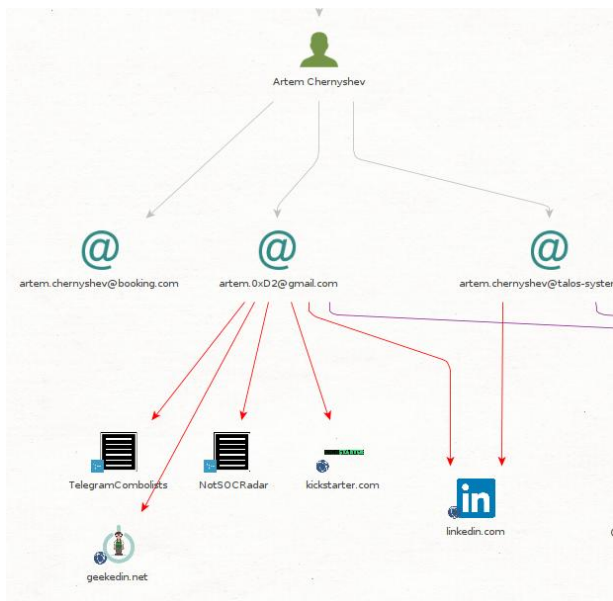


Aunque no disponemos del rol o función en la organización de cada figura descubierta, podemos identificar un perfil asociado a desarrollador, como lo es **Artem**.

Analizaremos los correos electrónicos asociados a esta persona (posible empleado de la organización) en <https://haveibeenpwned.com/>:

- artem.chernyshev@booking.com
- artem.0xD2@gmail.com
- artem.chernyshev@talos-systems.com

Podemos destacar las siguientes entidades asociadas vulneradas asociadas a dichos correos, comprometiendo información sensible.



LinkedIn Scraped and Faked Data (2023) (spam list): In November 2023, a post to a popular hacking forum alleged that millions of LinkedIn records had been scraped and leaked. On investigation, the data turned out to be a combination of legitimate data scraped from LinkedIn and email addresses constructed from impacted individuals' names.

Compromised data: Email addresses, Genders, Geographic locations, Job titles, Names, Professional skills, Social media profiles



GeekedIn: In August 2016, the technology recruitment site GeekedIn left a MongoDB database exposed and over 8M records were extracted by an unknown third party. The breached data was originally scraped from GitHub in violation of their terms of use and contained information exposed in public profiles, including over 1 million members' email addresses. Full details on the incident (including how impacted members can see their leaked data) are covered in the blog post on [8 million GitHub profiles were leaked from GeekedIn's MongoDB - here's how to see yours](#).

Compromised data: Email addresses, Geographic locations, Names, Professional skills, Usernames, Years of professional experience

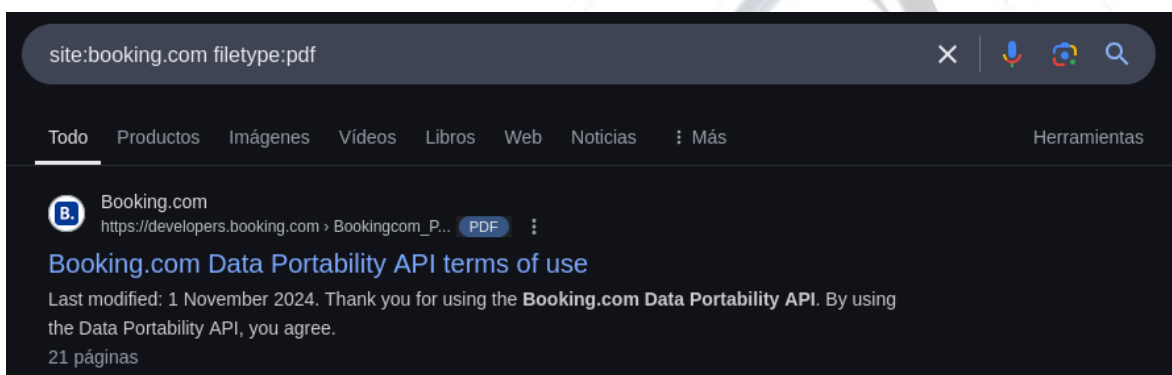
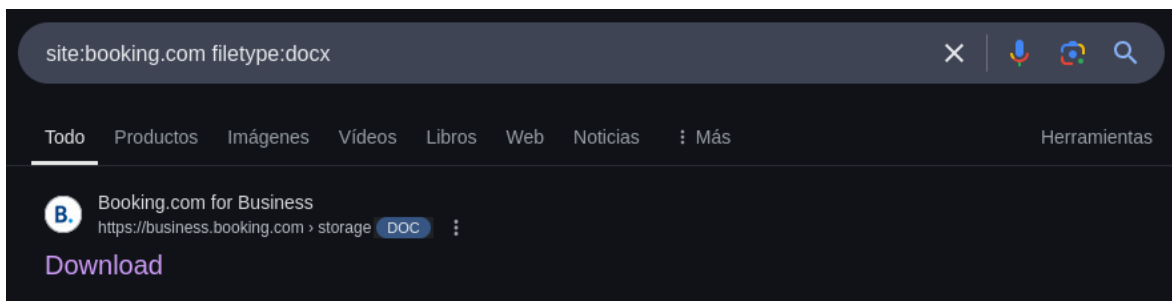


DemandScience by Pure Incubation: In early 2024, a large corpus of data from DemandScience (a company owned by Pure Incubation), appeared for sale on a popular hacking forum. Later attributed to a leak from a decommissioned legacy system, the breach contained extensive data that was largely business contact information aggregated from public sources. Specifically, the data included 122M unique corporate email addresses, physical addresses, phone numbers, employers and job titles. It also included names and for many individuals, a link to their LinkedIn profile.

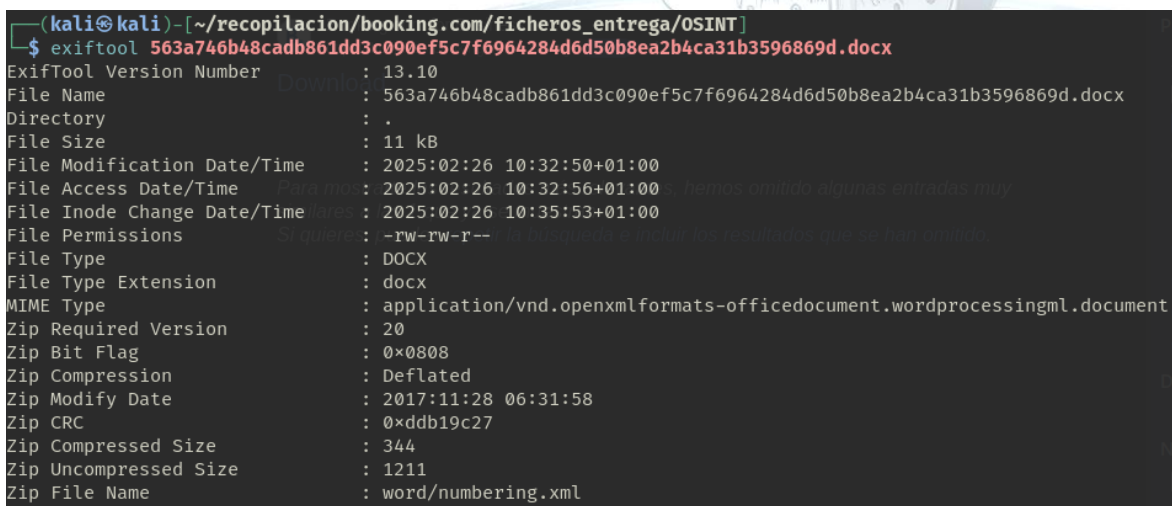
Compromised data: Email addresses, Employers, Job titles, Names, Phone numbers, Physical addresses, Social media profiles

Este tipo de información es importante para conocer el comportamiento interno y los procedimientos conocidos por los empleados de la organización ante escenarios de phishing.

Avanzaremos con una búsqueda de fichero compartidos expuestos en internet de diferentes formatos conocidos, como docx, xlsx, ppt, pdf:



Analizaremos los archivos descargados con **exiftool**




```

(kali㉿kali)-[~/recopilacion/booking.com/ficheros_entrega/OSINT]
$ exiftool Bookingcom_Portability_API_Terms_of_Use_v1.pdf
ExifTool Version Number      : 13.10
File Name                    : Bookingcom_Portability_API_Terms_of_Use_v1.pdf
Directory                    : .
File Size                    : 373 kB
File Modification Date/Time  : 2025:02:26 13:12:35+01:00
File Access Date/Time       : 2025:02:26 13:14:33+01:00
File Inode Change Date/Time  : 2025:02:26 13:14:33+01:00
File Permissions             : -rw-rw-r--
File Type                    : PDF
File Type Extension          : pdf
MIME Type                    : application/pdf
PDF Version                  : 1.4
Linearized                   : No
Page Count                   : 21
Title                       : Booking.com Portability API Terms of Use.docx
Producer                     : Skia/PDF m132 Google Docs Renderer

```

Los datos resultantes no aportan información trascendente, ya que no hay información de versiones o usuarios.

Utilizaremos la plataforma **Spiderfoot** para un nuevo análisis del dominio:

Booking RUNNING

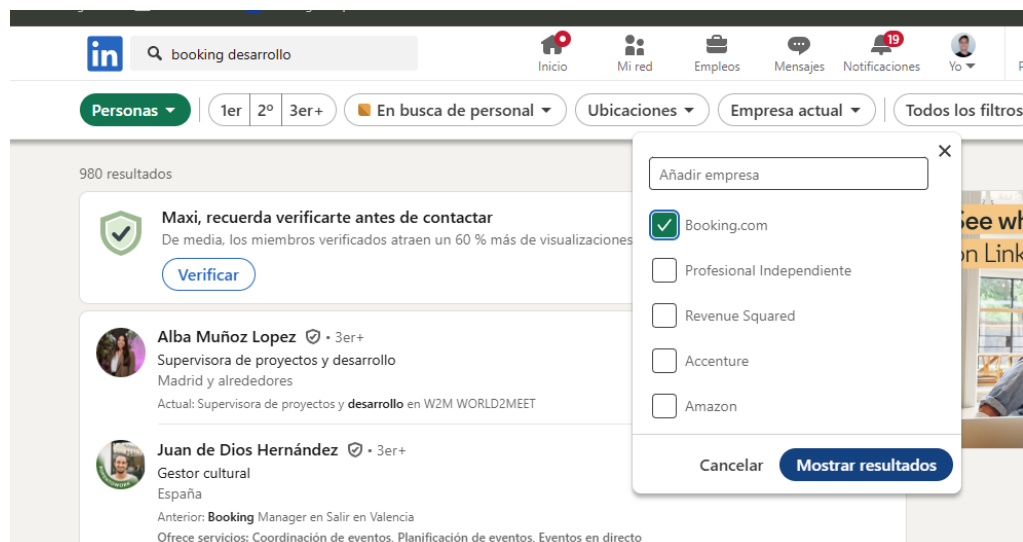
Summary Correlations Browse Graph Scan Settings Log

Search...

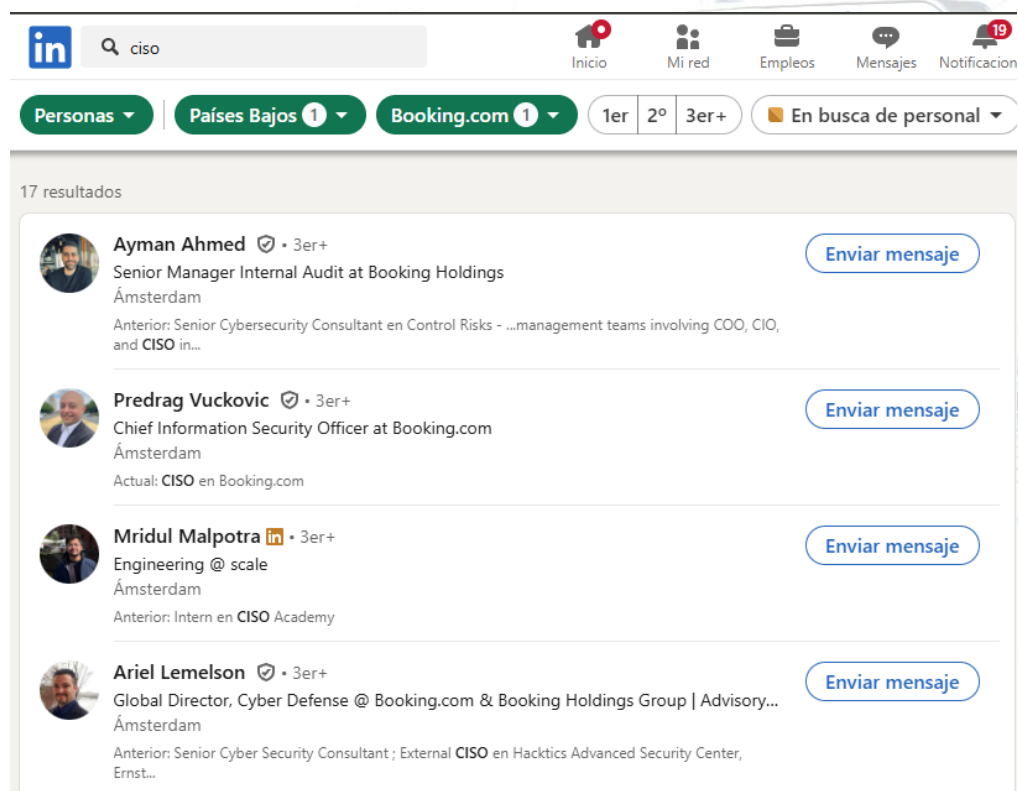
Type	Unique Data Elements	Total Data Elements	Last Data Element
Affiliate - Internet Name	8	8	2025-02-27 10:34:48
App Store Entry	1	1	2025-02-27 10:28:34
Cloud Storage Bucket	7	7	2025-02-27 10:34:46
Co-Hosted Site	95	172	2025-02-27 11:01:57
Co-Hosted Site - Domain Name	65	127	2025-02-27 11:01:09
Country Name	1	2	2025-02-27 10:30:37
DNS SRV Record	2	2	2025-02-27 10:29:56
DNS TXT Record	6	6	2025-02-27 10:34:47
Domain Name	1	4	2025-02-27 11:03:59
Email Address	62	63	2025-02-27 10:32:59

Los datos analizados en "Email Address" no muestran indicios de mail pertenecientes a roles importantes o puestos claves.

Haremos una búsqueda de posibles empleados en redes sociales, nos enfocaremos en la red **LinkedIn**. Utilizaremos los siguientes parámetros de búsqueda y exploraremos los resultados:



Conociendo el origen de la organización, Países Bajos, modificaremos los parámetros en busca de información:



LinkedIn search results for "seguridad".

138 resultados

- Naman Sharma** • 3er+
Information Security Professional at Booking.com
Amsterdam Area
[Enviar mensaje](#)
- Erin Giarratano** • 3er+
Trust and Safety Analyst / Human Trafficking Lead @ Booking.com | Human Rights
Ámsterdam
[Enviar mensaje](#)
- Cristian Rotari** • 3er+
Software Development Engineer at Booking.com
Ámsterdam
[Enviar mensaje](#)
- Shivam Shinde** • 3er+
Workday Functional | Workday Reporting & Analytics | Workday Technical
Países Bajos
[Enviar mensaje](#)

LinkedIn search results for "finance".

Aproximadamente 1.000 resultados

- Juan Negri** • 3er+
Finance Specialist
Países Bajos
Actual: **Finance** Specialist en Booking.com
[Enviar mensaje](#)
- Alexandros Baier** • 3er+
Finance Professional
Róterdam
Actual: Sr **Finance** Business Partner en Booking.com
[Enviar mensaje](#)
- Miembro de LinkedIn**
Strategic Finance (FPA) at Booking.com
Ámsterdam
- Miembro de LinkedIn**
Finance Specialist | MSc Finance | Payment Operations| Lean Six Sigma Green Belt | Scrum Master
Países Bajos

RESUMEN

Objetivo

El informe pretende dejar en evidencia las herramientas y conocimientos adquiridos en el módulo, gestionando con diferentes técnicas de recopilación sobre un dominio definido.

En cada apartado se ha generado observaciones con una mirada crítica sobre posibles vulnerabilidades.

Herramientas

Utilizamos las siguientes herramientas para explorar la información:

- Kali
- Shuffledns
- Analyticsrelationships
- Cero
- Katana
- CTFR
- Gau
- Alterx
- Dnsx
- Httpx
- Masscan
- Nmap
- Gowitness
- Wappalyzer
- Wafw00f
- Ffuf
- Greenbone
- Nuclei
- Wpscan
- Qualys -web-
- dmarcian.com -web-
- Subzy
- Maltego
- Spiderfoot
- Exiftool
- LinkedIn -web-

Se adjunta al informe el directorio “booking.com” y los respectivos ficheros de cada análisis.